Postel

A NETWORK/440 PROTOCOL CONCEPT

Network Working Group
Request for Comments #187
NIC #7131
Categories: C3, C4, C5, C6,D7
Update: None
Obsoletes: None

Douglas B. McKay
Donald P. Karp

IBM Thomas J. Watson Research Center
Yorktown Heights, New York

recvd. 27 July 71

This RFC is being circulated as an information RFC. Its intent is to convey some of the thinking and philosophy that went into IBM's network protocol and overall network design.

INTRODUCTION

Network/440 is an experimental project in computer netting that was undertaken by the Computer Science Department of IBM Research. The primary objectives of the project have been to understand netting, identify design problems and implement the solutions to these problems.

The above objectives have been met since a network has been built and is presently being operated by the project. Implementation discussions transpired with another department at Research in order to define a realistic user system interface. The protocol defined for the project's network is also the basis for the operation of an IBM OS network.

The Network/440 project has also been involved in the philosophical and architectural concepts of network systems. The basic premise in our work is the concept of a logical network machine.[1] The main theme is to treat all systems involved in the network as a part of a single (large) multiprocessor system. Although many of the ideas have been based on hypothetical concepts, an equal number of ideas were derived from our network implementation and operating experience.

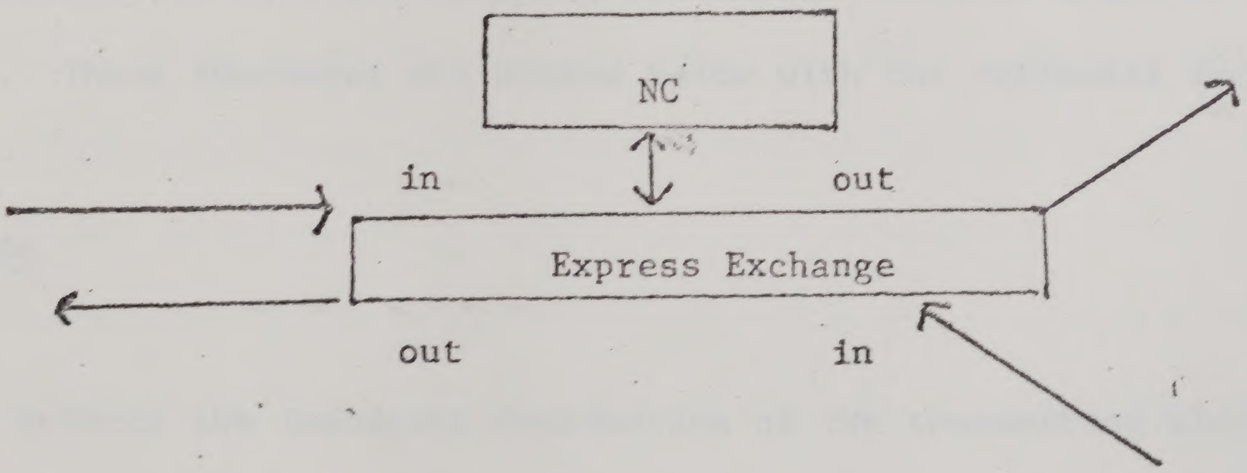
The scope of this paper is to describe the philosophy and definition of a network protocol that is not restricted to any physical configuration. This is exemploified by the fact that a major portion of the ideas are implemented in IBM's two major operational networks, one of which is a distributed configuration and the other a star configuration.

1. Intenet - Report 2, February 1, 1970, Computer Science Department, IBM Corporation, T. J. Watson Research Center, Yorktown Heights, New York.

BASIC ASSUMPTIONS

There was a necessity to delineate many network functions in setting up an operating protocol. These functions included switching control, buffer control, message control, and operating control. The operating control function becomes further complicated as the user is able to program the network as if it were a single operating system. The protocol had to be further broken down into detailed functions in order to cope with error recovery and handling techniques.

The original thoughts on handling these functions were to provide two basic realms of control. The net control is a higher level function that recognizes and controls all aspects of net jobs and the execution of job steps in the network machine. In addition, a communication control facility (referred to as an "Express Interpreter") was incorporated to provide fast service for all messages that were to be moved between user systems without intervention by the net controller.



The above figure illustrates the two major functions with messages travelling in both directions and directly through the Express Exchange, except in the case of messages that must be acted on by the Net Controller. These messages will be explained in detail later.

These two functions can exist on any system and operate in any physical configuration providing the control information reflects the configuration so that

proper operation can be maintained. There is no reference to physical configuration in this paper because of the flexible nature of the protocol and its adaptability to any configuration. For example, in the case of a distributed net, the Express Exchange would pass messages directly to the next station without any 'NC' overhead. The 'NC' would only come into play at the final destination and with the same reasoning, the 'NC' would not have to be present at every station.

DEFINITIONS

Before proceeding with the discussion of protocol and control, the basic message content and concepts must be defined.

A transmission block is a physical entity that consists of header and text. A message (logical) consists of many transmission blocks.

Header	Text

The primary purpose of the network is to deliver messages from one user system to another in an orderly controlled manner. In order to provide all the information necessary to maintain control, the header contains a set of operational functions. These functions are listed below with the rationale for each.

Action Code

This code selects the immediate destination of the transmitted blocks; the data may be transmitted directly to the user described in the DSID field, sent to 'NC', or used by 'EE'. Any conflict in information between this field and any other field in the header will cause an error message to be returned to the originating station. The AC will serve a similar function at the receiving system, indicating to the communications interface (CI) whether the

data block is destined for a user routine or contains control information for the CI. [The CI is that function which interfaces directly with the local operating system.]

Transmission Block Number

Each block of transmission within the network will contain a sequential number inserted by the transmitting station. As the block flows through the network, every station will insert its own number into the block, overlaying the previous station's number. The purpose of this sequential number is to guarantee that no messages are lost in the physical communications process.

Network Job Identifier

The function of this field is to associate a transmission block with the network job to which it belongs. The identifier is assigned to the network job and to each associated transmission block by the user system or by the 'NC'. In order to establish a unique name for each job within the network, the user node identifier (i.e., the name of the user system originating the net job) will be concatenated with a number generated by the originating user system.

Job Step (Marker)

The purpose here is to uniquely identify a job step within a network job. The NC will assign this name since it maintains control of all network jobs.

Originating System Identifier

In order to route a block of data from one user system to another, a unique name must be associated with each user system. The name will be assigned by the network control group at the time the user system is accepted as a network participant. The station originating a block of data will place his assigned identification in this field in every block of data originating at his system.

Message Priority

This field indicates transmission priority (not to be confused with processing priority) by block within the queue for a particular user system.

Destination System Identifier

This is similar to the originating node identifier except that the identification inserted is that of the node for which the block is destined.

Logical Message Flags

The message flags denote the first and last blocks of a message; all intermediate blocks are noted by their absence. The flag field in conjunction with the logical message sequence number will enable the user to determine if any blocks are missing from a message and will also provide an identifier that can be used to recover missing blocks. When the first and last indicators are turned on in a single block, the message is contained within the block.

Logical Message Sequence Number

This field is used to number sequentially the blocks within a message. The first block (denoted by the LMID) will contain the lowest number assigned (not necessarily 1) within a message while the last block will contain the highest number. Unlike the TBN, this number will remain intact throughout the journey of the block through the network. It is used for error detection and recovery along with the logical message flag.

Logical Message Identifier

Since all communications lines in the network can be multiplexed (blocks within a message will be interleaved with blocks from other messages), a message identifier becomes necessary in order to reassemble the message at the user destination. Therefore, each block within a message will contain an identifier unique to the message. In the simple case where the message is contained in one block, the identifier performs no function.

When multiple blocks comprise a message, LMID will enable the user to reassemble the message. There can be any number of physical message blocks associated with any logical message. It is important that the 'NC' assign numbers to this field for all messages it augments, and that this LMID be used in the messages generated by the CI in response to NC commands.

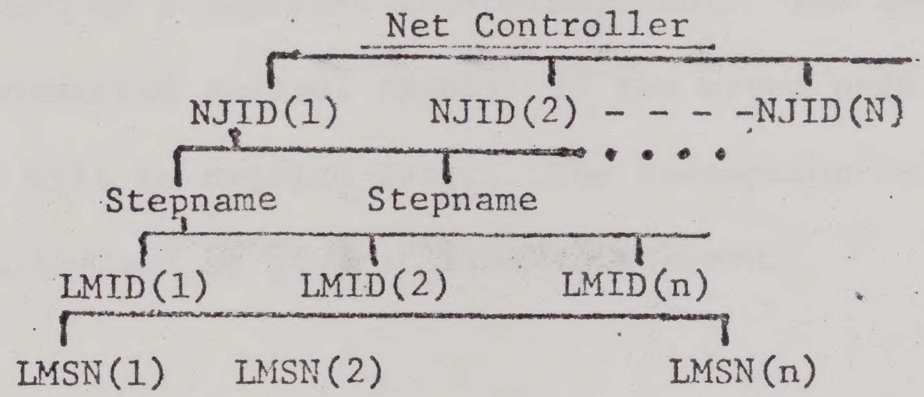
Length of Text

This field contains a binary number that equals the number of characters in

the text portion of the transmission block. Although there are other means available to obtain this number, it is included in the header for redundancy check purposes.

Logical Message Structuring

The network controller maintains control for every user job submitted by NJID. The following hierarchical structure is set up for a message configuration. Any message pertaining to any step in a network job can be tracked and retransmitted if necessary. It provides a mapping of the logical structure of any network job into their appropriate message configuration.



The Express Exchange is a combination of functions. It is basically a communication handler and store and forward switch. The 'EE' has the ability to keep track of all messages in the network by TBN(defined earlier).
It is therefore possible to record and reflect the entire status of the network down to any detail desired.

PROTOCOL

The protocol for operating a network system has different levels of control. The 'EE' must exercise control on the communication link between any pair of

stations. The NC maintains control at the net job level. However, the functions that each unit performs are combined to handle special control cases. These complimentary functions will be discussed in detail as they arise in the protocol discussion.

First of all, there must be a series of initialization messages sent from one station to another before any actual message transmission takes place. These messages are sent between each station and positive acknowledgments must be received in order to complete the initial hand shaking.

At any point during the transmission of messages an error can occur which will be detected by a negative acknowledgement. The message in error will be retransmitted several times. If the error persists, the line is timed out and will be retried later. The assumption here is the line may be temporarily noisy and we give it time to quiesce.

When a station receives an initialization message it is possible to respond in several ways depending on the status of the user system.

(1) The station receiving the initialization message can acknowledge that it is ready to receive and transmit.

(2) Temporarily cannot receive certain logical messages (actual data transmissions) but can receive special control messages. This option allows a user system to selectively process net jobs as facilities on his system become available.

(3) Unable to receive traffic (in other words, the user system is logically or physically disconnected from the network).

(4) Unable to receive new network job requests but able to handle traffic

for jobs in progress. The user system may have several jobs in progress that are transmitting and receiving messages. This acknowledgement gives the user system the ability to allow these jobs to continue normal processing.

The last alternative gives the CI at each user system the mechanism to selectively demultiplex itself to handling one logical message. The 'CI' also can suspend message transmission for any task that is temporarily deactivated.

Thus, all user systems can selectively halt messages throughout the entire network. The destination system can selectively halt all messages for a given NJID or selective halt logical messages within a net job. The adjacent system would keep accepting messages until its buffers were filled to some operational threshold limit that must be maintained to keep the network from coming to a complete standstill, and would issue selective halts to systems sending to it. It is conceivable that the message blocks of one logical message would be stored in distributed segments throughout the network.

The same selective halt mechanism can be applied in reverse through a resume message. The resume message can apply to an entire set of messages for a net job or selective logical messages within a job. The reinitiation of a transmission takes place between any two stations that wish to allow more message blocks to be transmitted. The destination station must resume on a particular logical message to allow the message to reach its final destination and complete transmission through the network. The LMID of the message header enables the 'EE' and 'NC' to cooperate in controlling and cleaning up network operation. Not only does this cooperation between logical levels reduce a duplication of effort but it enables the control to become realistic and practical. Complete

separation of communications and control functions could cause a loss of useful information that may not be obtained by other means.

For example, if a file transmission consisted of many blocks and a transmission error occurred that the network was unable to recover. The 'EE' would notify the 'NC' of the error occurrence on this file transmission and then 'NC' would issue purge messages to the 'EE's for those particular 'logic message' blocks. This mechanism allows a general 'clean-up' and management of all file transmissions.

There is also the condition when a receiving system goes down. When this occurs there may be a number of network jobs involved with that user system. If the user system remains down for an extended period of time and the 'EE' buffer resources are filled to threshold limit, it may be necessary to purge pending message blocks. The 'EE' will notify the 'NC' of the user system being down and the 'NC' will issue purge commands to the 'EE' for all pending messages of those netjobs involved with the down user system.
However, in our present implementation the 'EE' uses disk storage as a logical extension of core for message buffering. In this operation, the freeing of real core buffers becomes a simple matter of moving the messages on to disk for later retrieval. In some instances of transmission a file may be stored in segments at several locations until the receiving system is able to receive it. Network buffer resources are treated as a logically simple entity that may be physically distributed.

When the user system comes back on the air the involved user network job will be restarted by issuing resume transmit commands to the 'EE'. If the

user is an interactive user controlling the network, he would be notifed of the problem and status of his file transmission. He could then reinstate his command at a later time. The batch network job would be restarted at a point where no unnecessary retransmission would occur.

It has not been determined how long files should reside in a store and forward node before being purged from the network. If a backing storage device is available to network operation, the file can remain for a longer time but still not indefinitely.

NC PROTOCOL

The File Transmission Protocol of the 'NC' is primarily concerned with the control and transfer of user files for storage, temporary use at a remote system, and execution.

The commands and status messages that pertain to the second level logic of the 'NC' are sent and interpreted by the sending and receiving systems. All initiation of file transfers result from direct user commands to the 'NC'.

The sending system will first be interrogated to determine if the file is resident at that system. The user must provide the necessary information to locate the file if it is not catalogued at that system. This information consists of the physical attributes, such as volume and serial number. A negative acknowledgement to this message would result in the termination of a net job step with the reason for termination returned to the originator.

When a positive acknowledgement is received by the 'NC' it has two options available. It must first determine the amount of unused buffer space in the 'EE' and based on the size of the file to be transferred, decide whether to have the data set sent immediately or wait for an acknowledgement to the receive message.

If the 'NC' decides to move the file regardless of the state of the receiving system, the 'NC' will issue a send or receive message to both systems simultaneously. A negative response to the 'receive' message is taken as a definite refusal by the receiving system to accept the data transmission. This may result from insufficient resources to handle the job. If the file was transmitted from the receiving system and is resident in the network storage facilities, the user will be notified of its exact location so that he may move it from that point at a later time. If the 'NC' chose the second option, the file would still be resident at the originating system.

A positive acknowledgement will allow the file to continue its normal flow through the network. Queuing in the 'EE' is always done in order that 'receive' messages will be sent before the actual data file. The 'receive' message will specify what is to be done with the file. These possibilities include loading the file directly into the job stream (this step assumes the appropriate JCL is included in the text of the files) or cataloguing the file at the remote system or storing it for temporary immediate use. All network files are catalogues with a unique name that includes User ID (unique at his home node), home node ID (unique in the network) and his own data name which is unique in his own work. The 'receive' message may also contain some special instructions to print or punch a file.

When a negative acknowledgement is received by the 'RC' it has two options available. It then first determines the amount of network buffer space in the 'RC' and based on the size of the file to be transmitted, decide whether to have the data set sent immediately or wait for an acknowledgement in the response message.

If the 'RC' decides to send the file regardless of the state of the receiving system, the 'RC' will issue a sending network message to that network location number. A negative response in the 'receive' message, it serves as a definite refusal on the receiving system to accept the data transmission. This may be made since insufficient resources to handle the data. If the file was transmitted from the receiving system and is resident in the network storage facilities, the user will be notified of its exact location so that he may choose to pick that up at a later time. If the 'RC' chose the second option, the file would still be resident at the originating system.

A positive acknowledgement will allow the file to continue the normal flow through the network. However, if the 'RC' is always sent an answer that 'receive' message and in fact follows the actual data file, the 'receive' message will specify what is to be done with the file. These possibilities include, sending the file directly into the job stream. (Just make sure the appropriate JCL is contained in the body of the files), or cataloguing the file at the remote system or storing it for temporary immediate use. All network files are catalogued with a unique name that includes that of [illegible] at his home [illegible], have [illegible] (unique to the network) and his own data name which is unique in his own [illegible] the 'receive' message may also contain some special instructions to print or punch a file.

When the sending and receiving stations have completed the file transfer, they send status messages back to the 'NC' indicating the completed action. These status messages enable the 'NC' to keep a record of user network job steps and their progress through the network. These status messages play an important part in insuring proper checkpoint restart for the network.

Files routed specifically for execution require a third status message from the receiving user system. The system must indicate when and how the job completed execution. This statuss message will also contain the appropriate accounting information to allow dynamic updating of network user and system accounting information. It is not clear at this time what should be accounted for in the network, but it is an area of prime concern to operational networks.

An error in the second logic level can occur during the file transmission. There may be an error moving files from devices into the line buffers or reading from the line buffers. When this occurs, the operating system must pass this information to the 'NC'. The 'NC' will then terminate the task involved in this job step and purge all the network buffers containing blocks of this message transmission.

When the 'NC' receives the file error message it will immediately send a 'release' message to all the network tasks supporting this job step. This action will cause the user systems to end all pending tasks associated with this net job step. In addition a purge message for that job step will be sent to the 'EE' to purge the message from its buffers. If there is more than one 'EE' involved, the purge message would be passed to all other 'EE's.

This is another example of the 'EE' and 'NC' combining functional capability and providing effective management of network traffic. The mapping of message into the job step allows the 'NC' to selectively choose all messages it wishes to purge.

The protocol the user must use for interactive use of the network is different. There are some standard message types that are provided for interactive use to insure the proper message recognition from one system to another. Terminal type traffic will be sent across the network through the normal netting interface. The control information that a terminal sends to the operating system must be incorporated in the network protocol by the 'CI'.

The interactive user can request a direct connection to the remote system through the 'NC'. The 'NC' will notify the remote system of the user request and establish the user's direct link. The 'NC' becomes a monitor of the conversation but no longer becomes involved with the messages. Other conversational messages are sent back and forth through the 'EE' with no interaction by the 'NC'. In the event one of the systems goes down breaking the logical link, the 'NC' must notify the other system to terminate the waiting task. In most cases a user system will be isolated from the second user system by other stations and the 'NC' is a convenient way of notifying other user systems about the "disaster."

Once the user's connection is established, three types of messages may be generated. These messages are identified by the 'AC' field in the header. The three basic transmission types covered by the protocol are: a response

requested - with or without text included in the message, a text message which is simply a response to the first or just data to be printed at the user's terminal, and finally, an interrupt message which indicates the user wishes to stop a task or talk directly to the operating system.

It is important to note that regardless of what type of conditions exist, there are always enough buffers left to receive an interrupt message and terminate or flush any existing task and the associated operation it may be supporting.

CONCLUSION

The protocol concepts discussed in this paper were developed to facilitate the transfer of data between two or more independent systems. The protocol is able to handle the various pathological cases that may arise during network operation. A fundamental design consideration in developing these concepts was to maintain complete recovery from any recoverable error condition.

Many of the concepts have been used in an operational star network, with a single 'EE' and 'NC' located in the central system and a 'CI' located at each participating system. The successful operation of the network has proven the feasibility of this protocol.

ACKNOWLEDGMENT

The authors wish to acknowledge the design and implementation effort of the contributing members of the Computer Science Department of the T. J. Watson Research Center.